Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus

V. Ramakrishnan, Y. Zhuang, S. Y. Hu, J. P. Chen, C. C. Ko, B. M. Chen and K. C. Tan

Department of Electrical Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 117576

Abstract

The Internet provides an environment for developing several kinds of applications for educational purposes. This paper presents the implementation of a web-based laboratory experiment on a *Coupled Tank Apparatus*, a MIMO (Multi-Input Multi-Output) system. The web-based laboratory has been developed to serve undergraduate students and academic research staff in the Department of Electrical Engineering at NUS (National University of Singapore). It serves as an educational tool for teaching students the basic principles and methodology adopted in performing a series of experiments on a coupled tank apparatus. It also enables students to perform experiments at any time and from any location through the Internet. Additionally, it provides a platform for research staff to test control algorithms. The control strategies adopted include manual control, PID control, general state-space control and fuzzy logic control. The implementation uses video conferencing to provide audio and video feedback to the user, on the actual happenings in the laboratory and also allows the user to control the zoom and viewing angle of the video. This web-based remote laboratory can be accessed at http://vlab.ee.nus.edu.sg/vlab/control/.

Keywords : Virtual Laboratory; Internet; Web-Based Control.

1 Introduction

The concept of a web-based laboratory is not new. Shor and Bhandari [2] developed a distance learning application that allowed a user to remotely conduct experiments in the Control Engineering Laboratory at Oregon State University. This permitted the experiments to be run via the WWW (World Wide Web). Access required only a basic web browser that ran Java. A robot experiment was developed for demonstration. Bytronic Process Control unit [3], referred to as the process rig in the Process Control and Automation Laboratory at Case Western Reserve University, can be accessed remotely via the Internet. Using a web browser, the user can log on and post the parameters from a remote client to a LabVIEW (Laboratory Virtual Instrument Engineering Workbench from National Instruments [6]) web server that is connected to a process rig via a PLC (programmable logic control) module. The image is generated by a program and refreshed on the client side using server push technology.

A remote laboratory called VLAB [1] on an oscilloscope experiment was set up recently in the Department of Electrical Engineering, at NUS (the National University of Singapore). The existing remote laboratories such as [1], [2] and [3] are actual laboratory experiments that are run remotely via a web interface. These laboratories are well suited to distance learning courses where students need not be physically present on campus. The experimental parameters can be set on the web. A software interface converts those parameters to a form that is accepted by the local computer running the experiment. The core of all the Remote Laboratories is a cluster of general-purpose and/or specialised instruments interfaced to a set of personal computer systems connected to the Internet. These laboratories facilitate the sharing of expensive instruments and equipment, and represent the next important step in remote distance learning.

In this paper, the implementation of a remote laboratory control experiment on a coupled tank apparatus is presented. Four different types of commonly used control structures are implemented viz., manual control, PID control, general state-space control, and fuzzy logic control. This experiment is an industrial control related experiment for third year undergraduate students at NUS (the National University of Singapore). LabView [6] by National Instruments, Java programming language, HTML (Hyper Text Markup Language) as well as scripting languages like JavaScript, VBScript have been used in this implementation. A distinctive feature of the implementation is the use of video conferencing using Microsoft NetMeeting [8] for providing audio and video feedback to the user on the happenings in the laboratory. The laboratory set-up of the equipment used in this implementation is depicted in Fig. 1.

Fig. 1. Laboratory Set-up of the Equipment

This paper is organised as follows. Section 2 describes the complete structure of the Virtual Laboratory including its hardware and software structures. It also explains the use of TCP for client-server communication in the implementation as opposed to CGI and the distinctive features of the implementation viz. audio feedback, video feedback and control of

the pan/tilt and zoom of the camera. The model of the coupled tank is presented in Section 3. Section 4 presents the structures of the different controllers that are used and the user interfaces for performing the four types of control. The conclusion is made in Section 5.

2 Structure of the Virtual Laboratory

2.1 Hardware Structure

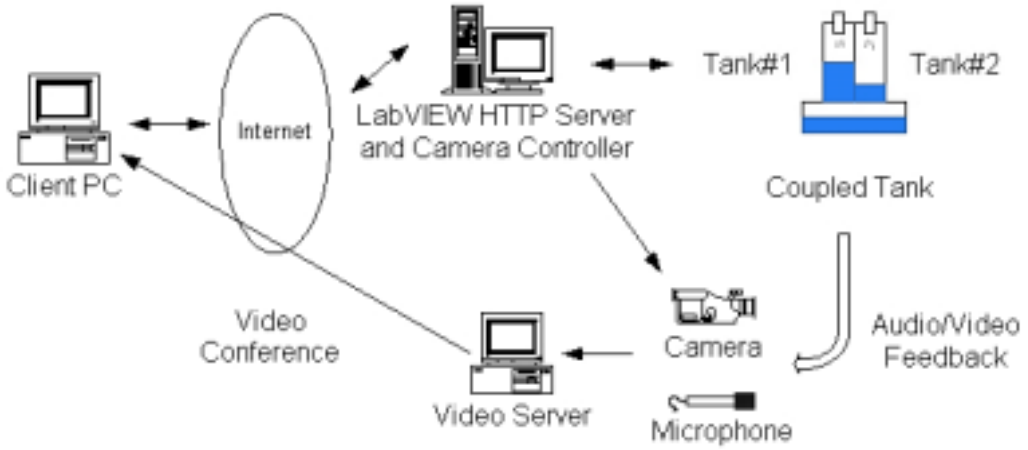

Fig. 2. Hardware Structure of the Virtual Laboratory

The hardware structure is shown in Fig. 2. As shown, a client PC connects to the HTTP server (provided with LabView [6] as part of its Internet Toolkit) which hosts the web site for conducting the experiment. The PC running LabView functions as the instrument controller. It is connected to the coupled tank through a 50-pin DAQ (Data Acquisition) card and implements the local control of the coupled tank apparatus by supplying the two input voltages to the coupled tank. A camera connected to a PC running Microsoft NetMeeting i.e. the video server, enables the provision of visual feedback to the user, on the changes in water levels in the coupled tank. A microphone connected to the same PC captures the sound of the coupled tank's motor and provides audio feedback to the user. The pan and tilt of the camera as well as its zoom setting are controlled by a circuit board supplied by American Dynamics [12]. This circuit is also connected to the instrument controller. The live image data captured by the camera and the sound captured by the microphone are sent to the client through a video

conference session. This is a point-to-point transfer implying that only one user can view the video at a time.

2.2 The Use of TCP for Client Server Communication

CGI and TCP are two popular methods for implementing client-server communication. CGI programs running on a web server sending and receiving parameters from the users' web browsers have become an Internet standard as depicted in Fig. 3.

The client uses an Internet browser to connect to the web server, view the web page, send the CGI request to the server side and finally receive the response form the server side. The CGI connection is based on HTTP. Since HTTP is a per-session-basis protocol, the connection between the client and the server is closed once the client receives the response from the server. As opposed to a CGI connection, a TCP connection established by a client with a server remains in place till the client closes the connection.

Since the establishment of a TCP connection implies that the connection will be in place till the client closes it, TCP is the ideal communication protocol for implementing the web-based control of instruments that require frequent parameter adjustments. CGI, on the other hand, is totally unsuitable for this purpose due to its operation on a per session basis. The communication between a client and server using TCP is as shown in Fig. 4.

Fig. 3. A Typical CGI Connection Process

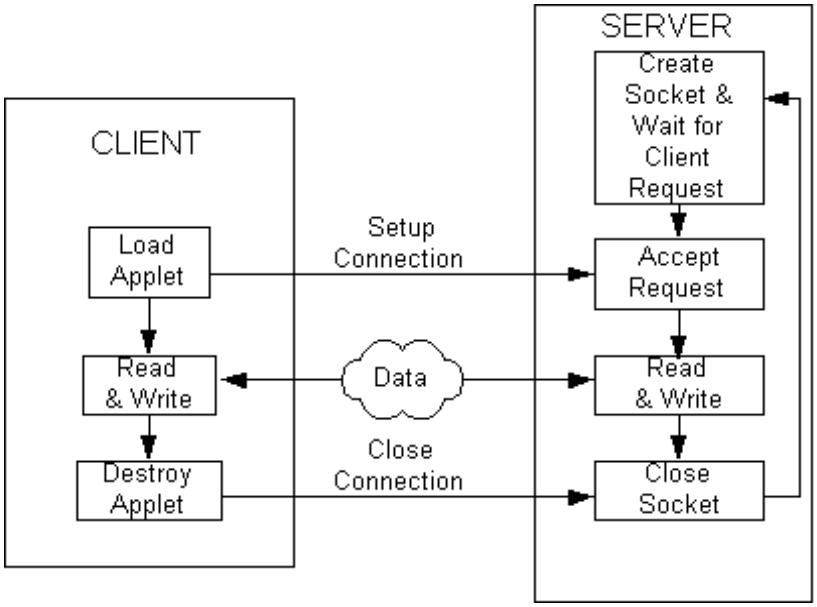

Fig. 4. Client-Server Interaction using TCP

The communication between the client and server can be set up conveniently using Java. Java programs are compiled to platform independent codes, called bytecodes, of a hypothetical machine called the JVM (Java Virtual Machine). These bytecodes can be viewed by a Java-enabled web browser. Using Java, the connection between the client side and the server side can be set up through sockets. When the user downloads a Java applet from the web server, the applet (running on the client machine) represents the client side of the connection. Java was preferred to other programming languages for implementing client-server communication due to its platform independent nature and the extensive network programming support.

2.3 Audio and Video Feedback

A striking feature of the implementation is the use of video conferencing and audio feedback as opposed to multicast schemes to provide the user with visual feedback. During the initial stages of development, real time streaming of image data was attempted using the

two most popular commercial solutions available at the time. These solutions involved the use of RealServer G2 [9] and NetShow [10] media servers to serve the live video data as well as the use of commercial decoders RealPlayer G2 and Windows Media Player at the client side. However, these solutions were observed to be unfit, as a time lag of the order of ten seconds was noticed between the actual occurrence of an event in the laboratory and its presentation at the client. This delay was caused predominantly due to software overheads. These overheads arose because the multicast schemes were optimised to support a large number of clients. The *Multi-Datarate Encoding* (based on the bandwidth available to the target audience), *Buffering* (during network congestion) and *Intelligent Transmission* (maintaining continuity of video at the expense of quality during network congestion) features of the multicast schemes were also identified as potential bottlenecks.

Since the campus network at the National University of Singapore, NUSNET-III [11] provides the user with bandwidth comparable with a T1 connection, bandwidth is not a constraint for the implementation of the remote laboratory experiment within the campus.

Video conferencing was thus adopted to provide live video to the client. For this purpose, Windows NetMeeting 3.0 was chosen. NetMeeting uses the H.323 standard, which is comprised of ITU (International Telecommunications Union) approved protocols for audio, video, and data conferencing over TCP/IP networks. NetMeeting uses the H.323 standard-based video technology, which is also complaint with the H.261 and H.263 video codecs.

NetMeeting 3.0 also supports the transfer of audio. A microphone connected to the instrument controller captures the sound made by the Coupled Tank's motor. The sound in turn can be heard at the client side by attaching speakers to the client computer. This feature is intended to provide the user a realistic feel while performing the experiment as though it is being performed in the actual laboratory. Volume control is effected by adjusting the speaker's volume control on the client side.

The implementation uses an ActiveX control to embed the live audio and video on web pages. The control is part of the Windows NetMeeting Resource Kit [13]. The ActiveX

control is invoked and controlled using VBScript, a scripting language developed by Microsoft Corporation. When the user activates the ActiveX control and calls the peer i.e. the video server, explicitly by specifying the IP (Internet Protocol) of the video server, a call is placed. The NetMeeting program on the video server is configured to receive calls immediately without any user intervention on the server side for accepting the call. It is also configured to send out live audio and video at the start of each call. The client thus receives live audio and video. The receipt of audio and video at the client side can be stopped in a similar manner thereby ending the call.

It is apparent in this scheme that only one user can receive video at a time. This is logical since only one user is allowed to access the web-based laboratory at a time. This single user feature is implemented by restricting access to the experiment's web pages. The user is authenticated by a CGI program residing on the HTTP server, and only then allowed to access the web pages for performing the experiment.

2.4 Camera Control Features

Yet another special feature of the implementation is the control of the camera pan and tilt as also its zoom. The camera is mounted on a pan and tilt unit, which can be controlled through a circuit. The circuit connects to the controller PC's serial port through an RS-232 connector. The controller PC can issue commands to control the circuit. The commands used for this purpose are described in Table 1. The circuit in turn can control the pan/tilt unit and camera zoom by applying appropriate voltages at the concerned terminals.

FUNCTION	ASCII	HEX	DEFINITION
Camera	#	**23**	Addresses the intended receiver/driver (camera). This command must be preceded by the camera number.
Aux Off	**B**	**42**	Turns an auxiliary off. This command must be preceded by the auxiliary number.
Aux On	**A**	**41**	Turns an auxiliary on. This command must

			be preceded by the auxiliary number.
Focus Far	**F**	**46**	Focus lens far.
Focus Near	**N**	**4E**	Focus lens near.
Iris Close	**C**	**43**	Close lens iris.
Iris Open	**O**	**4F**	Open lens iris.
Lens Wide	**W**	**57**	Zoom lens wide.
Lens Tele	**T**	**54**	Zoom lens telephoto.
Pan Left	**L**	**4C**	Pan camera left.
Pan Right	**R**	**52**	Pan camera right.
Tilt Up	**U**	**55**	Tilt camera up.
Tilt Down	**D**	**44**	Tilt camera down.
Call Shot	****	**5C**	Calls a preposition scene. This command must be preceded by a scene number.
Set Shot	**^**	**5E**	Sets (stores) a preposition scene. This command must be preceded by a scene number.

Table 1 : Commands for Camera Control Circuit

Web-based control is implemented by continuously running a camera control program on the machine housing the HTTP server, to receive command strings from the client (through a TCP channel) for the local control of the circuit. At the client end, a Java applet sets up the sockets for TCP communication with the server. When the user clicks a button intended for camera control, a method defined in the applet, that generates the corresponding command string and writes it to the TCP connection, is invoked.

2.5 Software Structure

This sub-section explains how the web-based control of the coupled tank is implemented. The PC running the LabView HTTP server also runs LabView 'G' language programs that set up the server side of the TCP communication with the client. The software structure of the system at runtime is as shown in Fig. 5. As shown, there are four Java applets, which run on the client and handle the necessary communication with the server and plot the response curve using the data obtained from the server. A fifth applet communicates

with a server program by sending it command strings for camera zoom and pan/tilt control. An ActiveX control executes on the client and receives audio and video data from the video server.

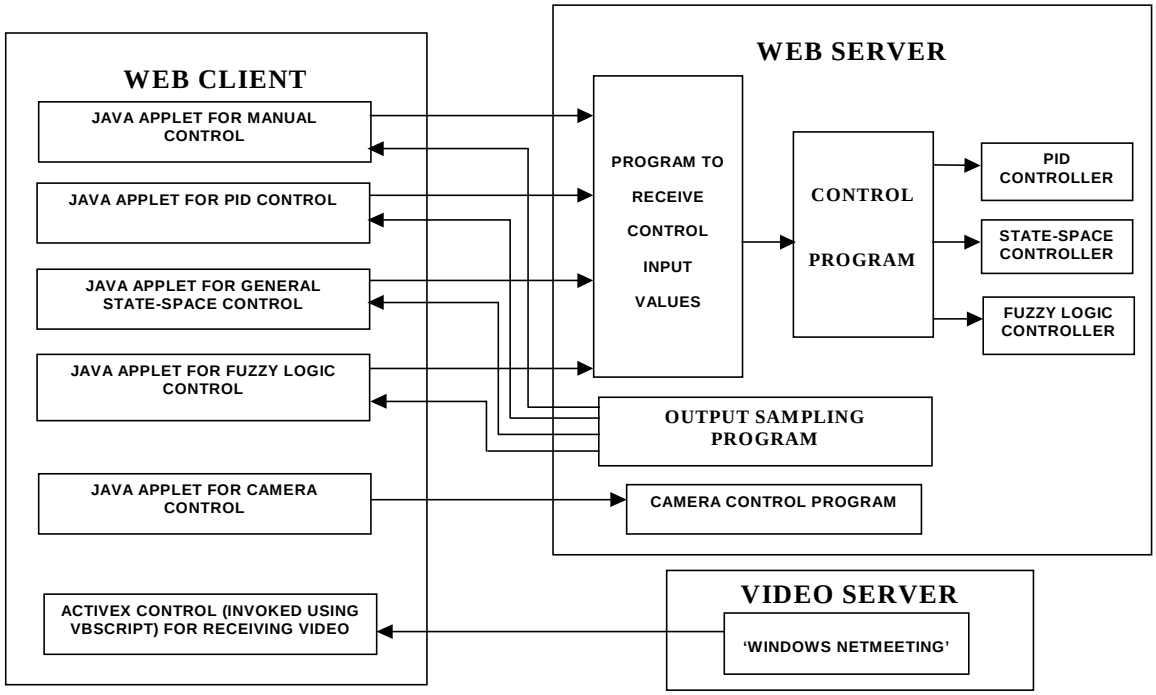

Fig. 5. Software Structure of the Virtual Laboratory

On the server side there are four programs to handle communication with the client and receive the controller parameters and the reference input (set point) values input by the user. These parameters are passed on to the controller modules (coded in LabView G) which implement the local control of the coupled tank. In the case of manual control however, the received voltage values are directly fed to the coupled tank. The program for controlling the mounting unit of the camera is coded in Visual Basic also resides on the same machine. The working of the LabView G programs for receiving the parameters from the user is now described.

LabView supports network communication protocols, such as TCP and UDP (User Datagram Protocol), implemented in the form of Virtual Instruments' sub VIs. LabView uses the following VIs to communicate through the Internet.

i. The TCP Listen.vi: This VI waits for an incoming TCP connection request. It also returns a connection ID when the TCP connection is created.

ii. The TCP Read.vi: This VI receives a specified maximum number of bytes to read from the specified TCP connection.

iii. The TCP Write.vi: This VI writes the string data to the specified TCP connection.

iv. The TCP Close Connection.vi: This VI is used to release the TCP connection.

The server side G programs therefore, receive the information about the two set points and the parameters for the various controllers and pass on this information to the two controller blocks (one corresponding to each input to the coupled tank). The output of each controller block is converted to an analogue voltage by the DAQ card and fed to the coupled tank. The two output analogue signals from the coupled tank are sampled and the voltage information is sent to the client side. At the client, the voltage information is then used to draw the curve corresponding to the system response by means of a Java applet.

3 Modelling the Coupled Tank Apparatus

We start by describing the model of the plant i.e. the coupled tank apparatus. To obtain the overall system response, the plant must be modelled first. Four different types of control structures, which are implemented, are described later in section 4.

The coupled tank apparatus (shown in Fig. 6.) is designed for the teaching of process control principles in engineering courses. It consists of two perspex tower-type tanks mounted above a reservoir which stores water. The head of water in the tanks can be read from the scale in front of the tank. Each tank is fitted with an outlet, which in turn is connected to a plastic hose for returning water to the reservoir. The outflow rate of water returning to t he reservoir, is approximately proportional to the head of water in the tank. The

level of water in the tanks is monitored by two capacitance probes, each of which along with some electronic circuits, provides an output signal proportional to the level of water in the corresponding tank. This output voltage is in the range of 0 to +10 volts. Complex tank arrangements are possible by varying the level of the internal baffle, which controls the inter-tank resistance. The controlling PC feeds two analogue input signals to the coupled tank through the DAQ card. The two analogue outputs corresponding to the two tanks are sampled at a specified rate (set by the user).

In the laboratory experiment chosen for implementation, the coupled tank is modelled as a two-input-two-output system by keeping the baffle raised and feeding in two analogue input signals and observing the two outputs. The non-linear and linearised state space models of the coupled tank are presented below.

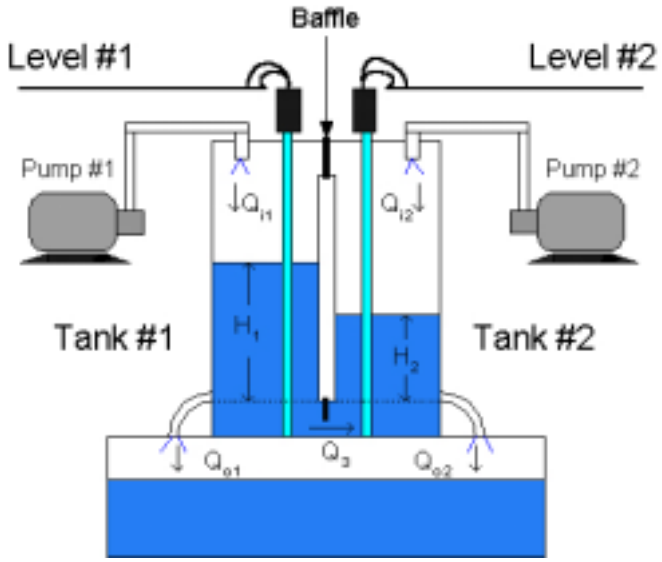

Fig. 6. Coupled Tank Apparatus

Given that,

H_1, H_2 = height of fluid in the tanks,

A_1, A_2 = cross-sectional areas of the tanks,

Q_{i1}, Q_{i2} = pump flow rate into tanks,

Q_{o1}, Q_{o2} = flow rate of fluid out of the tanks,

Q_3 = flow rate of fluid between the tanks,

and write down Bernoulli's equation for steady, non-viscous, incompressible flow which states that the outlet flow in each tank is proportional to the square root of the head of water in the tank –

$$Q_{o1} = \alpha_1 \sqrt{H_1} \quad \textbf{(1)}$$

$$Q_{o2} = \alpha_2 \sqrt{H_2} \quad \textbf{(2)}$$

Similarly, the flow between the two tanks is proportional to the square root of the head differential. Thus,

$$Q_3 = \alpha_3 \sqrt{H_1 - H_2} \quad \textbf{(3)}$$

The proportionality constants, α_1, α_2 and α_3 in the above equations are dependent on the coefficient of discharge, the cross sectional area of each orifice and the gravitational constant. The coupled tank apparatus with two inputs and two outputs and the baffle completely raised can be modelled in state space as explained below.

Combine equations (1) to (3) and observe Fig. 6, we can obtain the following set of non-linear, state-space equations, which describe the system dynamics of the coupled tank.

$$\begin{aligned} A_1 \frac{dH_1}{dt} &= Q_{i1} - \alpha_1 \sqrt{H_1} - \alpha_3 \operatorname{sgn}(H_1 - H_2) \sqrt{|H_1 - H_2|} \\ A_2 \frac{dH_2}{dt} &= Q_{i2} - \alpha_2 \sqrt{H_2} - \alpha_3 \operatorname{sgn}(H_2 - H_1) \sqrt{|H_1 - H_2|} \ . \end{aligned} \quad (4)$$

Furthermore, the dynamics of each pump connected to the each tank can be approximated as,

$$\begin{aligned} Q_1 &= k_1 u_1 + c_1 \\ Q_2 &= k_2 u_2 + c_2 \ . \end{aligned} \quad \textbf{(5)}$$

where Q_1 and Q_2 are inflow rates of tank1 and tank2 respectively, u_1 and u_2 are control signals for pump1 and pump2 respectively, k_1, c_1 and k_2, c_2 are constants of pump1 and pump2 respectively.

Therefore, if we control the coupled tank with the following pre-feedback law,

$$\begin{aligned} Q_{i1} &= [\alpha_1\sqrt{H_1} + \alpha_3 \operatorname{sgn}(H_1 - H_2)\sqrt{|H_1 - H_2|} - c_1] + (k_1u_1 + c_1) \\ Q_{i2} &= [\alpha_2\sqrt{H_2} + \alpha_3 \operatorname{sgn}(H_2 - H_1)\sqrt{|H_1 - H_2|} - c_2] + (k_2u_2 + c_2) \end{aligned} \quad \textbf{(6)}$$

the system will be simplified as

$$\begin{aligned} \frac{dH_1}{dt} &= \frac{k_1}{A_1}u_1 \\ \frac{dH_2}{dt} &= \frac{k_2}{A_2}u_2 \ . \end{aligned} \quad \textbf{(7)}$$

4 Controlling the Coupled Tank Apparatus

Four different types of control which have been implemented on the coupled tank are now described.

4.1 Manual Control of the Coupled Tank

Manual control consists of directly feeding the control input values to each tank without the use of any automatic controller, but that of user. The user is expected to adjust the values of the control inputs to achieve the desired control. The user interface provided for manual control is shown in Fig. 7. We note that this structure can be used to determine critical frequencies of the process, which would in turn provide useful information on selecting various gains in PID control.

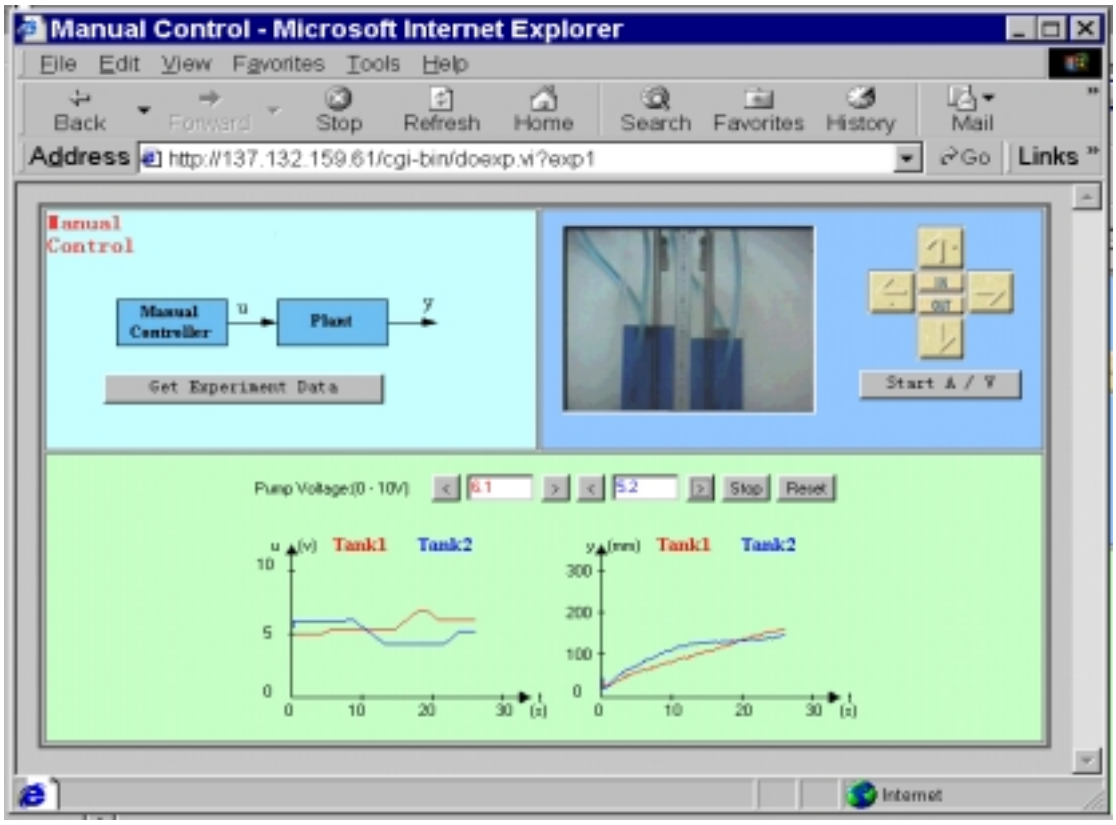

Fig. 7. User Interface for Manual Control

4.2 A PID Controller for Coupled Tank Control

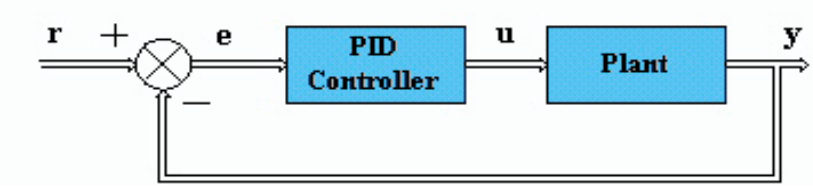

Fig. 8. Block Diagram of Experimental System Using a PID Controller

The block diagram of the overall system comprised of the PID controller and the plant i.e. the coupled tank, is shown in Fig. 8. Here, *r* refers to the reference input viz. the set point. The continuous time response of a PID controller is given by

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \qquad \textbf{(8)}$$

where K_p, K_i and K_d are respectively the proportional, integral and derivative gains. In the implementation of the coupled tank experiment, the PID controller has been realised through a software program and therefore, the appropriate discrete time response is

$$u(k) = K_p e(k) + K_i T_S \sum_{j=0}^{k} e(j) + K_d \frac{e(k) - e(k-1)}{T_S} \qquad \textbf{(9)}$$

While performing PID control, the user is expected to input the values of the two reference

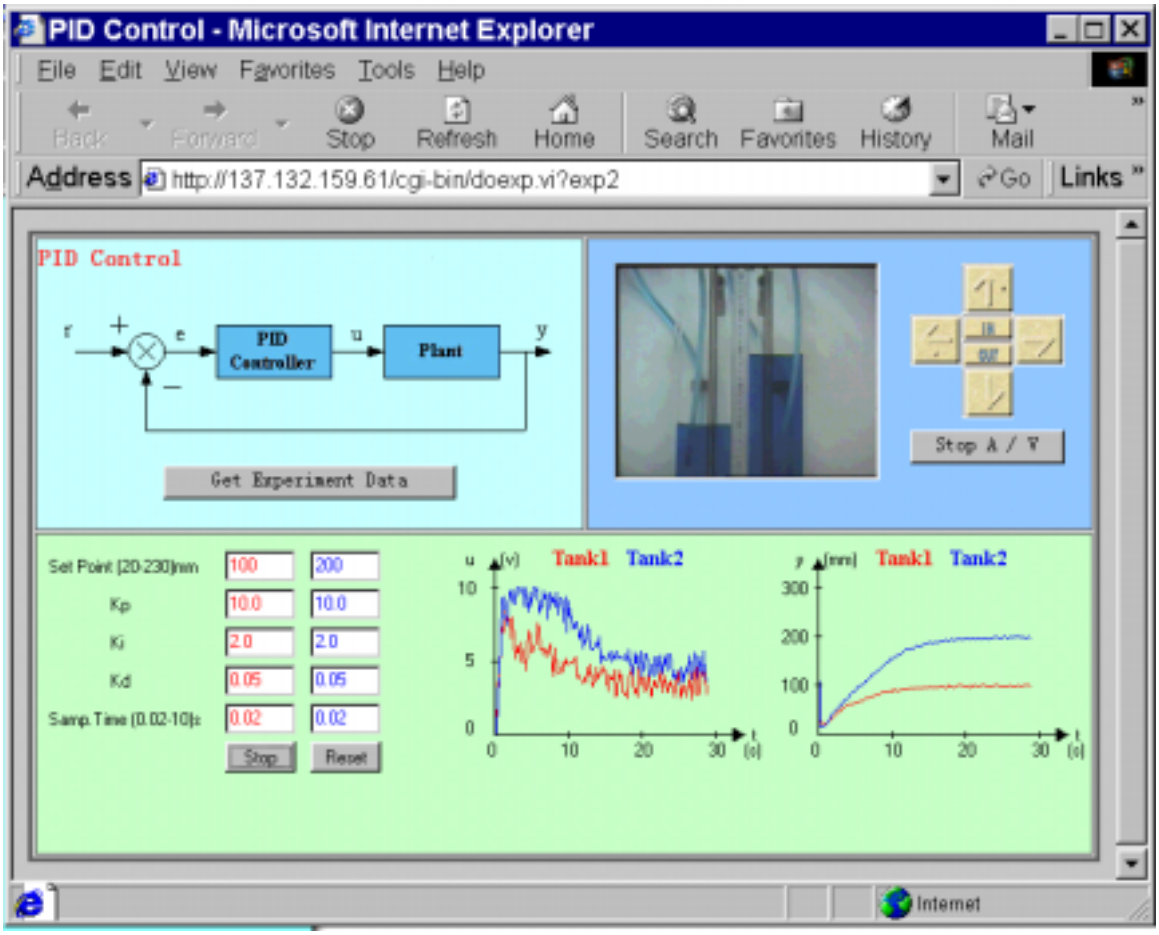

Fig. 9. User Interface for PID Control

inputs (set points) and the values of the proportional, integral and differential gains. These values are obtained by tuning the controller using various established procedures. The user interface for PID control is shown in Fig. 9.

4.3 Control using a Generalised State Space Controller

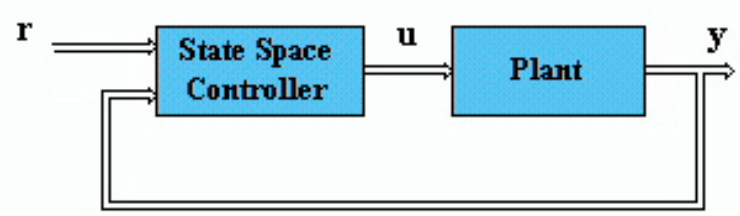

Fig. 10. Block Diagram of Experimental System Using a State-Space Controller

The block diagram for the control of the coupled tank using a generalised controller modelled in state space is illustrated in Fig. 10. Here, $\boldsymbol{r}$ is the refrerence input, $\boldsymbol{y}$ is the measured output of the coupled-tank and and $\boldsymbol{u}$ is the control input. The state model of the controller is then specified by the equations

$$\dot{\boldsymbol{x}}_C = \mathrm{A}_C \boldsymbol{x}_C + \mathrm{B}_C \boldsymbol{y} + \mathrm{G}_C \boldsymbol{r} \qquad \textbf{(10)}$$

$$\boldsymbol{u} = \mathrm{C}_C \boldsymbol{x}_C + \mathrm{D}_C \boldsymbol{y} + \mathrm{H}_C \boldsymbol{r} \qquad \textbf{(11)}$$

where $\boldsymbol{x}_C$ is the state of the controller, and A_C, C_C, D_C, H_C, B_C and G_C are some appropriate dimensional constant matrices. The user interface for performing general state-space control is shown in Fig. 11. In conducting experiment using this general state-space controller structure, the user is first prompted to enter the order of the controller that he or she has designed. Based on this value, a user interface will automatically be generated for the user to enter data as shown in Fig. 12 (for a case when the order is specified as 5). The data will be sent to the software controller on the server side once the user clicks the *OK* button.

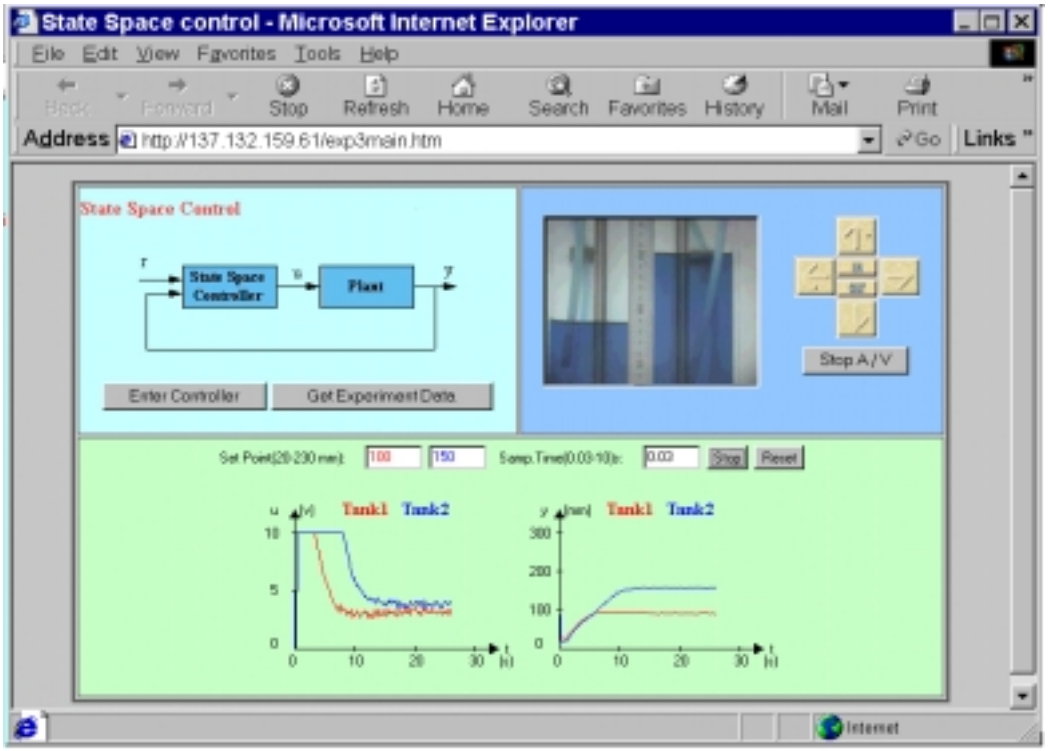

Fig. 11. User Interface for General State-Space Control

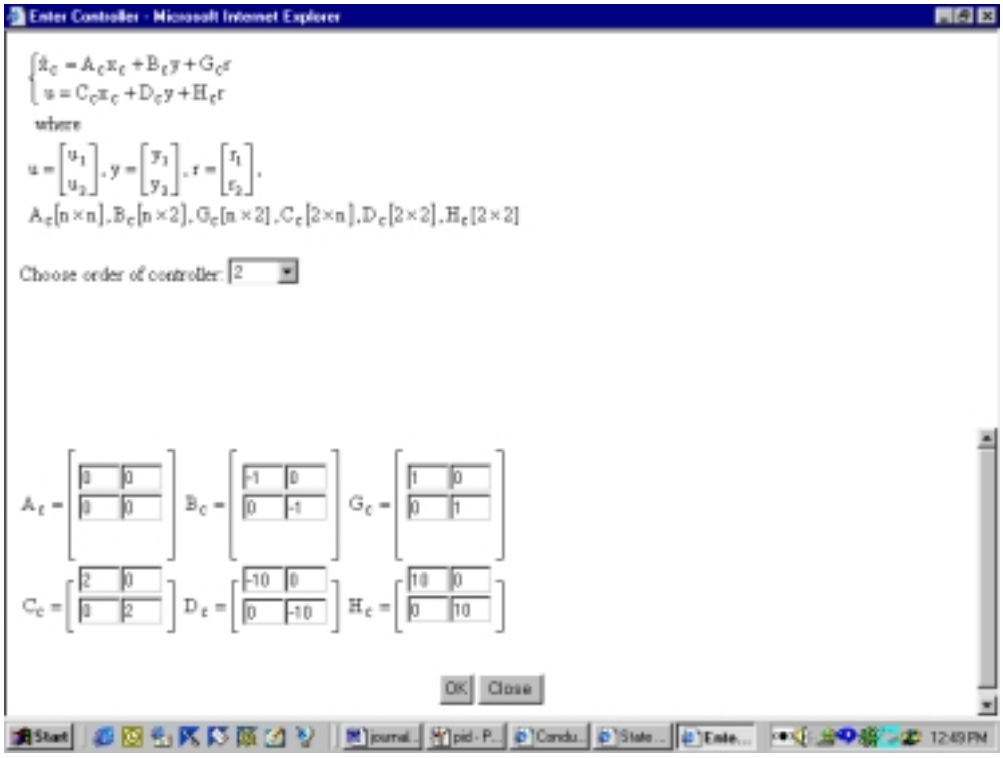

Fig. 12. User Interface for Input of Matrices

4.4 A Fuzzy Knowledge-Based Controller

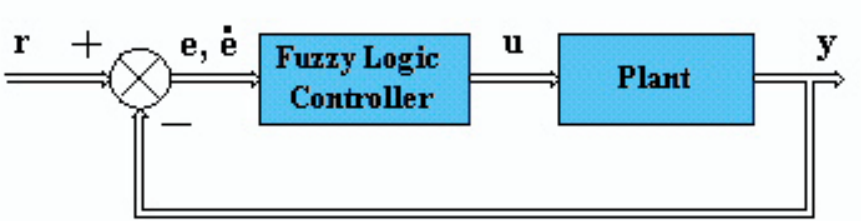

Fig. 13. Block Diagram of Experimental System Using a Fuzzy Logic controller

A FKBC (Fuzzy Knowledge-Based Controller) can be developed to implement the control of the coupled tank. The system block diagram in this case will be as depicted in Fig. 13. The principal components of a FKBC are illustrated in Fig. 14.

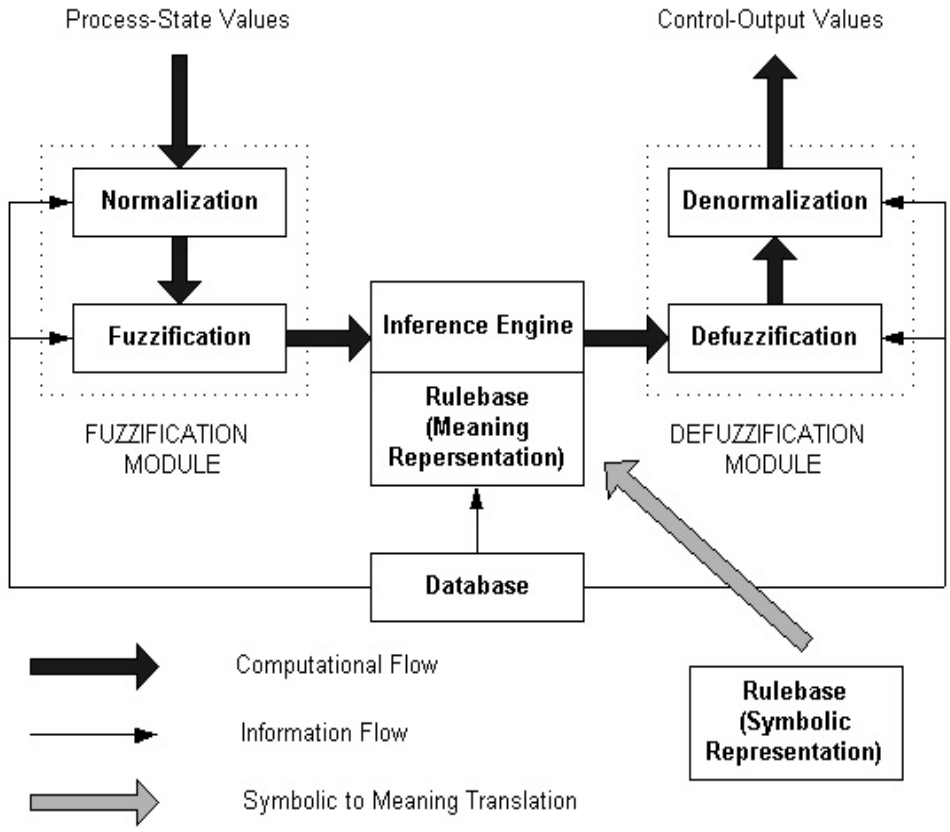

Fig. 14. The Principal Components of a Fuzzy Knowledge-Based Controller

As shown, the three principal blocks of a FKBC are the fuzzification module, an inference engine and a defuzzification module. The fuzzification module includes a normalization block for mapping the physical values of the control state variables onto the normalised domain. This block also maps the normalised control output variable onto its physical domain. The knowledge base of the fuzzy controller is composed of a database and a rule base. The database includes information about the membership functions and also about the normalization/denormalization scaling factors. The rule base represents the control policy adopted. For the implementation of fuzzy control on the Coupled Tank, the process state variables are the error (e) and the change-of-error ($\dot{e}$). The control output variable is the control output (u). The term sets of $e, \dot{e}$ and u have all been chosen equal and contain the same linguistic values. Thus,

$$Lu = Le = L\dot{e} = \{\text{NL, ZR, PL}\}$$

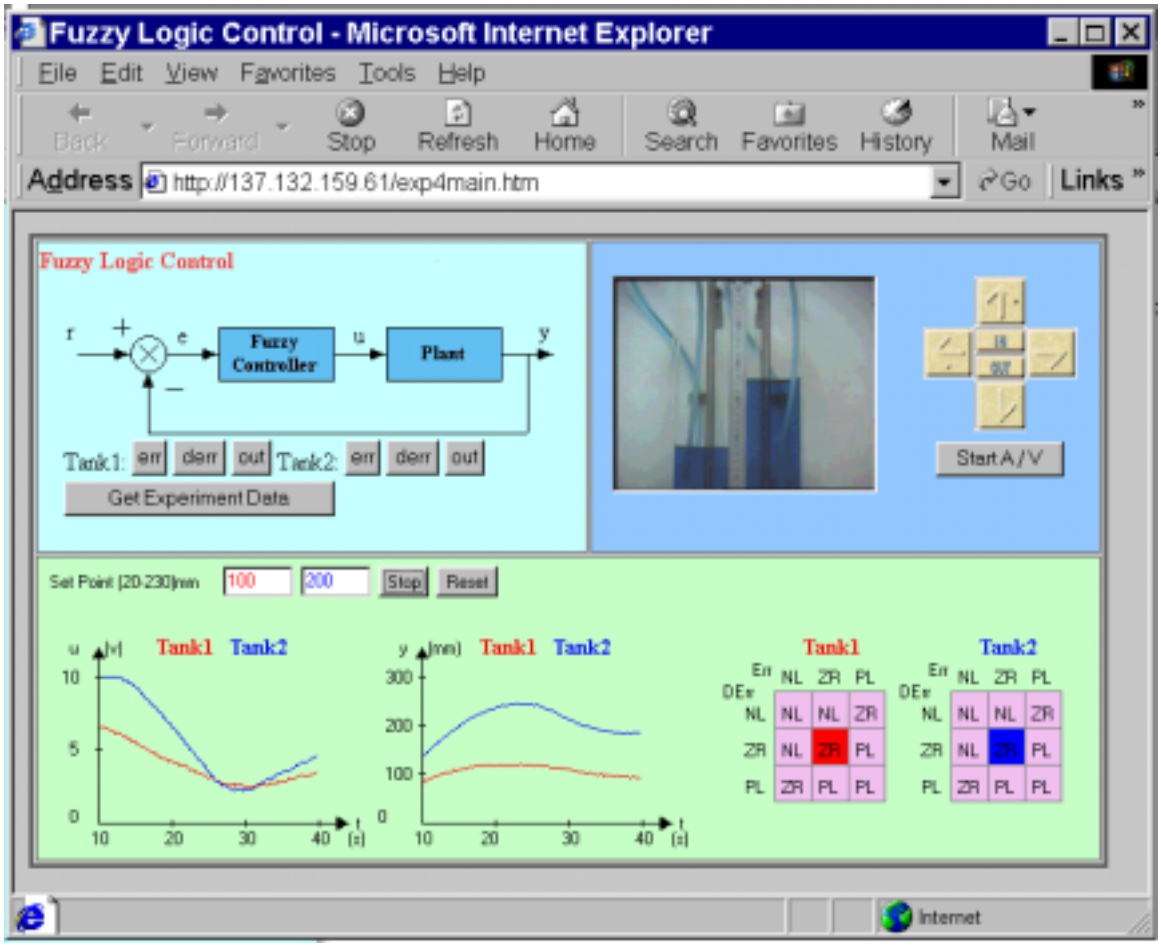

Fig. 15. User Interface of Fuzzy Knowledge-Based Control

The user interface developed for fuzzy logic control is shown in Fig. 15. The default rule base shown in two tables at the bottom right corner can be modified by the user by clicking on the table entries. The membership functions for error, derivative of error and output can be modified by clicking on the *err*, *derr* and *out* buttons respectively. On clicking these buttons a new window pops up as shown in Fig. 16 to enable the user to edit the membership function.

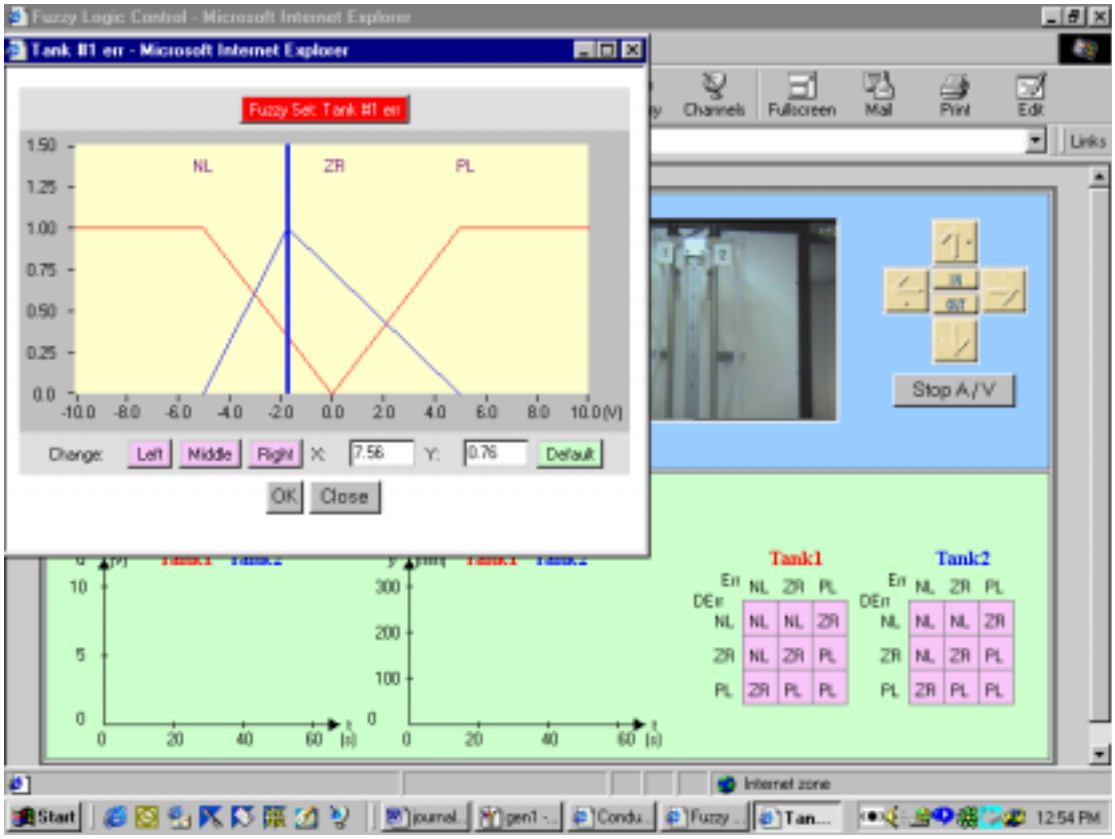

Fig. 16. User Interface for Editing Membership Functions

By clicking on the *Left*, *Middle* and *Right* buttons the user can choose the function to edit. A vertical line then appears which can then be dragged using the mouse to the desired position. The default membership functions can be restored by pressing the *Default* button.

The inference engine has been realised to use individual rule based inference. The normalization/ denormalization scaling factors used are 10.0, 50.0 and 1.0 for $e, \dot{e}$ and u respectively. Currently, efforts are on to develop and implement a learning algorithm for the rule base, membership functions and scaling factors.

5 Conclusion

Based on the methodology described in [1], for the development of remote laboratory experimentation through Internet, a web-based laboratory experiment on a Coupled Tank Apparatus was implemented with some new and attractive features. These features are the

use of video conferencing for providing audio, video feedback to the user and the provision for adjustment of the pan/tilt and zoom of the camera capturing the real-time image. The laboratory experiment intended to serve undergraduate students and academic research staff in the Department of Electrical Engineering at NUS, has been mounted on the web and can be accessed at http://vlab.ee.nus.edu.sg/vlab/control/. The experiment is concerned with the control of a Coupled Tank Apparatus through manual control, a PID controller, a general state-space controller as well as a fuzzy controller. Current efforts are directed towards developing a learning algorithm for the fuzzy logic controller.

References

[1] S. H. Chen, V. Ramakrishnan, R. Chen, S. Y. Hu, Y. Zhuang, C. C. Ko and B. M. Chen, "A large scale web-based virtual oscilloscope laboratory experiment," Engineering Science and Education Journal, in press.

[2] Shor, M. and Bhandari, A., "Access to an Instructional Control Laboratory Experiment Through the World Wide Web", *Proc. of the 1998 American Control Conference*, Philadelphia, pp. 1319-1325, 1998.

[3] M. Shaheen, K. A. Loparo and M. R. Buchner, "Remote Laboratory Experimentation", *Proc. of the 1998 American Control Conference*, Philadelphia, pp. 1326-1329, 1998.

[4] *Web-based Virtual Laboratory* at National University of Singapore, http://vlab.ee.nus.edu.sg/vlab.

[5] S. H. Chen, R. Chen, V. Ramakrishnan, S. Y. Hu, Y. Zhuang, C. C. Ko, B. M. Chen, "Development of Remote Laboratory Experimentation through Internet", *Proceedings of the 1999 IEEE Hong Kong Symposium on Robotics and Control*, Hong Kong, pp.756-760, July 1999.

[6] *LabView, User Manual*, National Instruments, 1998.

[7] *Coupled Tank Control Apparatus, Model: PP-100, Operator and Service Manual*, Kent Ridge Instruments, Singapore, 1999.

[8] *Windows NetMeeting 3.0 Help*, Microsoft Corporation, 1999.

[9] *RealServer G2 Help,* RealNetworks, 1999.

[10] *NetShow 3.0 Documentation*, Microsoft Corporation, 1999.

[11] *Home page of NUSNET-III* (http://www.cc.nus.edu.sg/)

[12] *Model 1641 Series Economical Receivers with Modular Code Board, Installation and Operating Instructions Manual,* American Dynamics, 1998.

[13] *Window NetMeeting Resource Kit,* Microsoft Corporation, 1999.